IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES
Attorney Docket No. 003797.87364

In re U.S. Patent Application of Jerry Dunietz, et al.	)	
Application No. 09/552,262	)	Group Art Unit: 2176
Filed: April 19, 2000	)	Examiner: Peter J. Smith
For: PRE-COMPUTING AND ENCODING TECHNIQUES FOR AN ELECTRONIC DOCUMENT TO IMPROVE RUN-TIME PROCESSING	)	Confirmation No. 4106

BRIEF ON APPEAL

Mail Stop: Appeal Brief-Patents
Commissioner of Patents
P.O. Box 1450
Alexandria, VA 22313-1450

This is an Appeal Brief in accordance with 37 CFR §41.37, filed in support of Applicant's April 24, 2006 Notice of Appeal. Appeal is taken from the Final Office Action mailed December 23, 2005, and the Advisory Action mailed March 10, 2006. Please charge any necessary fees in connection with this Appeal Brief to our Deposit Account No. 19-0733.

I. Real Parties in Interest

The owner of this application and real party in interest is Microsoft Corporation.

II. Related Appeals and Interferences

There are no related appeals and interferences.

III. Status of the Claims

Claims 1-35 are finally rejected. No claim is allowed. Claims 1-35 are pending and are being appealed herein. All of the pending claims, claims 1-35 are shown in the attached appendix. Claims 1, 10, 12, 14, 16, 19, 24 and 32 are independent claims.

IV. Status of Amendments

There are no amendments subsequent to the Final Office Action of December 23, 2005, and all prior amendments have been entered.

V. Summary of the Claimed Subject Matter

In making reference herein to various portions of the specification and drawings in order to explain the claimed invention, Applicants do not intend to limit the claims; all references to the specification and drawings are illustrative unless otherwise explicitly stated.

Aspects of the invention are directed to a process that converts an e-book from a general format, such as Open eBook, to a simplified file format hierarchy. (Pg. 5, Ln. 2-4). Typically, the general format includes tags that are commands written between "<" and ">" symbols, where the commands affect how the content is displayed and the tags can be heavily intermixed with the content. (Pg. 2, Ln. 25 – Pg. 3, Ln. 4). During the process, the conversion pre-computes and encodes the e-book to accelerate run-time search operations and to minimize computation requirements for run-time parsing and other forms of processing. (Pg. 5, Ln. 4-7). In an embodiment, the content and tags are separated and the tags are replaced with pre-defined integer representations. (Pg. 5, Ln. 17-19). Flags may be inserted into the tags to identify start and end tags, word breaks and area of content that should be skipped during a run-time search. (Pg. 5, Ln. 19 – Pg. 6, Ln. 2). For example, content associated with a tag containing the flag "NOSEARCH" indicates that content is hidden and need not be searched at run time. (Pg. 22, Ln. 3-6).

The converted e-book may be a single file within a nest hierarchy of files, where the nested files can be multiple directories and files. (Pg. 13, Ln. 1-15). Thus, the converted e-book has a root directory and linked metadata files as well as a linked content subdirectory within the root directory. (Pg. 13, Ln. 19-25). The root directory may include a linked manifest file that provides a list of all the files in the e-book so that the list does not need to be derived from the metadata. (Pg. 15, Ln. 21-22).

During encoding, a code character is inserted to separate markup language from the actual content, where the code character may be UNICODE character 0x0000. (Pg. 16, Ln. 1-3). The code character is inserted before and after each start and end tag. (Pg. 16, Ln. 3-4). Flags such as "NOSEARCH" and "WORDBREAK" may be inserted in between the boundary code characters to help identify the content associated with the tag. (Pg. 19, Ln. 19-22).

For example, the tag, which could be the tag "<td nowrap>" becomes the structure:

(1) 0x000

(2) Flags = STARTTAG | WORDBREAK = 0x000C

(3) Tag code = TAGID_TD = 95 = 0x5f

(4) Attribute code=DISPID_IHTMLTABLECELL_NOWRAP = 0x8001138D

(5) Attribute value = Boolean TRUE

(6) 0x000

(Pg. 23, Ln. 5-21). The sequence may then be encoded into Unicode characters, converted into UTF-8 format and compressed so as to minimize computational requirements during run time. (Pg. 24, Ln. 19 – Pg. 26, Ln. 5).

VI. Grounds of Rejection to be Reviewed on Appeal

Claims 1-2, 5, 7, 9, and 16-21 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Tada et al. (hereinafter "Tada"), U.S. Patent No. 5,745,745 patented 4/28/1998, in view of Fontaine, et al. (hereinafter "Fontaine"), U.S. Patent No. 5,228,121 patented 7/13/1993.

Claims 3-4, 8, and 22-23 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Tada, in view of Fontaine, and further in view of "Open eBook Publication Structure 1.0" (hereinafter "Open eBook") published 9/16/99.

Claim 6 stands rejected under 35 U.S.C. § 103(a) as being unpatentable over Tada in view of Fontaine, and further in view of Carus, et al., (hereinafter "Carus") U.S. Patent No. 6,035,268.

Claims 10 and 11 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Tada in view of Carus.

Claims 12 and 13 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Tada.

Claims 14 and 15 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Open eBook.

Claims 24-35 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Open eBook in view of Tada.

The rejections of claims 1-35 are being appealed.

VII. Argument

A. Claims 1-2, 5, 7, 9, and 16-21 are patentably distinguishable over Tada in view of Fontaine.

Claims 1, 16, and 19 are independent claims and stand rejected under 35 U.S.C. § 103(a) over Tada in view of Fontaine. Claims 2, 5, 7, and 9 depend directly from independent claim 1. Claims 17-18 depend directly from independent claim 16, whereas, claims 20-21 depend directly from independent claim 19. As will be shown below, the combination of Tada and Fontaine fails to disclose or suggest at least one claimed feature of all the rejected claims. Therefore, Tada in combination with Fontaine fails to support a *prima facie* case of obviousness for claims 1-2, 5, 7, 9, and 16-21.

1. Independent claim 1 is patentable because neither Tada nor Fontaine disclose or suggest the claimed feature "separating the tag from the content with a separation variable" recited in claim 1.

Tada discloses a text search method for structured documents. The Final Office Action states and Applicants agree that ". . . Tada does not teach combining the alias and the flag and separating the combination from the content with a separation variable." (Emphasis Added; Final Office Action, Page 3.) The Final Office Action relies on *Fontaine* in an attempt to disclose this missing claimed separation variable, stating:

> Fontaine does teach combining two or more tags or information objects into a single nested structure in col. 4 lines 31 – col. 5 line 44 and col. 5 line [sic]. Since, the nested structures of Fontaine logically define the order of a document and the informational objects contained within the document, the boundaries of the combined structure encapsulating the alias and flag of Tada would have been separation variables separating the encode tag structure from the content.

Final Office Action, page 3.

Therefore, as may be seen from the above cited passage, the Examiner is equating the boundaries of the combined structure to Applicants claimed "separation variable."

However, Applicants respectfully disagree, as combining tags and information objects does not disclose the claimed feature of "separating the tag from the content with a separation variable." The boundaries of a structure may not be equated with the insertion of a separation

variable. In fact, if the boundaries of *Fontaine* were discernable then the need for a separation variable would not be necessary.

In addition to and different from the boundaries argument discussed by the Examiner on Page 3 of the Final Office Action, the Examiner on pages 15-16 of the Final Office Action offers a specific example from *Fontaine* in which the Examiner alleges that the symbols %% used in the *Fontaine* preferred embodiment represent Applicants claimed separation variable. In particular, the Examiner states:

> Fontaine teaches an[d] [sic] example tag named "tagname" in col. 4 line 59. Since this tag cannot be distinguished from the content, a separation variable is plainly shown also in col. 4 line 59. In the example of Fontaine, the variable %% is used to separate the tag from the text content. The separation variable enables Fontaine to identify the tag and invoke its definition which is described in col. 4 lines 49-55. Therefore, the Examiner maintains the position that Fontaine does teach the claimed separating a tag from content with a separation variable.

Furthermore, the Advisory Action states:

> The Examiner believes that Fontaine teaches inserting a separation variable within the broadest reasonable interpretation of the claimed separation variable in col. 4 lines 49-55. These are variable characters and they are separating the tag from the content, and therefore reads upon the "separation variable" under the broadest reasonable interpretation.
>
> Advisory Action, Page 2.

Applicants respectfully submit that the instructions symbolized by %% symbols used in Fontaine do not disclose, teach, or suggest the claimed feature of "separating the tag from the content with a separation variable." In particular, Fontaine at Column 4, lines 49-55 states:

> The tag definition is initiated with an instruction ".tag tagname". Following the tagname, the contents of the information object or tag is stored. Once the tag definition is complete, the instruction ".endtag" is entered. The tag contents may then be retrieved through any desired compiler or retrieval implementation. For example, in the preferred embodiment the information (tag) is recalled through the instruction "%%tagname%%".

As may be seen in the cited portion of Fontaine, an information object (tag) is recalled through instruction "%%tagname%%." The symbols %% used in Fontaine refers to a recall instruction and do not disclose, teach, or suggest Applicants claimed "separation variable."

Support for Applicants claimed separation variable may be found on at least page 17 of Applicants specification:

> At step 410, a code character is inserted to separate markup language from the actual content of the e-book file. For example, the code may be a Unicode character 0x0000. The Unicode character is inserted before and after each start and end tag. Subsequent encoding of the markup (discusses herein) should be constrained so that the Unicode character 0x0000 never occurs within the representation of a start or end-tag or elsewhere within the content, but rather occurs only as a first and last character of each start an end tag.

As shown in the above cited portion of Applicants specification, the separation variable may be a code character such as Unicode character 0x0000. In addition, the separation variable is inserted before and after each start and end tag. As illustrated in above cited portion of the specification, the start and end tags are distinct entities from the claimed separation variable as the separation variable separates the tags from the content. Applicants submit that Fontaine does not disclose both the use of tags and a separation variable, the separation variable used to separate tags from content.

As neither Tada nor Fontaine disclose the claimed feature of "separating the tag from the content with a separation variable," independent claim 1 is patentable over the combination for at least this reason.

2. Independent claim 1 is patentable because neither Tada nor Fontaine disclose or suggest the claimed feature of "replacing the tag with an alias, wherein the alias is a pre-defined representation for the tag" of independent claim 1.

In addition, claim 1 is allowable over Tada and Fontaine for at least an additional reason. The combination of Tada and Fontaine fails to teach at least the claimed element of "replacing the tag with an alias, wherein the alias is a pre-defined representation for the tag" as recited in step (b) of claim 1. The Final Office Action relies on Tada, contending that "Tada teaches replacing the tag with an alias in col. 22 lines 6-20." Final Office Action, page 3. Col. 22 lines 13-28 of Tada states:

> A logical structure identification number corresponding to the logical structure discriminator is obtained from the correspondence table . . . . Specifically, in the example . . . the start tag '<Title>' is detected, . . . and the logical structure identification number '1' is obtained . . . .

> . . . At the search database creation step 35, in place of the start tag, a specific control code "α" representative of the start of the logical structure is written and the obtained logical structure identification number and logical structure length are written after the control code

At no point in the above description does Tada describe "replacing the tag with an alias, wherein the alias is a pre-defined representation for the tag" as recited in the claim. Rather, this portion of Tada merely describes obtaining an identification number, the identification number replacing logical structure discriminator (such as "a character string after the start character "<" and before the end character ">"). Col. 22, lines 10-11. Applicants respectfully submit that identification number in Tada is replacing content such as "a character string" and is not replacing a tag with an alias, the alias being a pre-defined representation for the tag.

Moreover, as stated above in Tada, a start tag is replaced by a control code "α." Applicant contends that the control code "α" is representative of a start of a logical structure and does not disclose an alias (ex. pre-defined integer) as claimed. Therefore, Applicants respectfully submit that replacing a start tag with control code does not disclose, teach, or suggest, the claimed feature of "replacing the tag, with an alias, wherein the alias is a pre-defined representation for the tag." As neither Tada nor Fontaine does this claimed feature, Applicants submit that for at least this additional reason independent claim 1 is patentable over the combination.

3. Independent claim 1 is patentable because neither Tada nor Fontaine disclose or suggest the claimed feature "inserting at least one flag within the tag to form an encode tag structure" recited in claim 1.

Furthermore, claim 1 is allowable over Tada and Fontaine for at least an additional reason. The combination of Tada and Fontaine fails to teach or suggest "inserting at least one flag within the tag to form an encode tag structure" as recited in step (c) of claim 1. The Final Office Action relies on Tada, contending that "Tada teaches in col. 22 line 24 – col. 23 line 24 inserting a control code, which is a flag, to form an encoded structure indicating whether the information contained within the tags should be searched or not." Final Office Action, page 3. Col. 22 lines 24-29 states:

> [I]n place of the start tag, a specific control code "α" representative of the start of the logical structure is written and the obtained logical structure identification number and logical structure length are written after the control code "α". The

end tag is deleted, and the control codes such as text ID and eot are written for the creation of the search database.

More explicitly, col. 22 lines 36-37 state "the start tag is replaced by the control code α . . . ." Therefore, Tada teaches replacing the tag with the control code. If the control code is a flag as the Final Office Action claims, then Tada does not teach or suggest "inserting at least one flag within the tag to form an encode tag structure" as recited in the claim. Rather, this portion of Tada would teach replacing the tag with the flag. Therefore for at least this additional reason, Applicants respectfully submit that claim 1 is in condition for allowance.

4. Independent claim 1 is patentable because the rationale in the Final Office Action for combining Tada and Fontaine does not set forth a convincing line of reasoning as to why a person of ordinary skill in the art would be motivated to modify the system disclosed in Tada by Fontaine as proposed in the Final Office Action.

The Final Office Action on page 3 states:

> It would have been obvious and desirable to have combined the alias and flag of Tada into a single structure in the form of a document object as taught by Fontaine so that the information would have retained a document format as is taught by Fontaine as opposed to the database format explicitly taught by Tada. Thus, by retaining a document format, the content could be manipulated and used as a document.

Moreover, the Final Office Action on page 16 states:

> Tada uses the created structure to improve the search of text content, as is a similar goal of the invention of claim 1. Fontaine teaches an improvement for separating a take from text context. The improvement of Fontaine would have enabled the tag search improvement of Tada to have been implemented in the document. Fontaine teaches placing and separating the tags within the text content directly in col. 4 lines 57-63. Thus, the Examiner believes this improvement to have been created by one of ordinary skill in the art at the time of the invention by expressly using the stated advantages of Tada and Fontaine.

In addition, the Advisory Action on page 2 states:

> Both Tada and Fontaine are directed to improvements in electronic document manipulation and therefore would have been known to one of ordinary skill in the art at the time of the invention. Therefore, it would have been obvious to have combined the advantages of Tada and Fontaine to improved the structure of electronic documents.

Applicants respectfully submit that the offered motivation to combine the teaching of Tada and Fontaine cannot be found in either Tada or Fontaine or in the knowledge of one of ordinary skill in the art at the time of the invention. It is respectfully submitted that just because the teaching of different documents may be combined and implemented together, it may not be inferred that such motivation existed at the time of the invention by those of ordinary skill in the art. Applicants respectfully submit that the Examiner is using the present application as a blueprint to combine references when the only suggestion can be found in the present application. Tada concerns a search method for searching structured documents in a massive structured document database; whereas, Fontaine is concerned with generating documents from information objects from one or more master files and sources files. Though Applicant agrees that both Tada and Fontaine utilize electronic documents it is unclear how one skilled in the art would combine their teaching to disclose Applicants claimed invention.

Therefore, for at least the reasons describe above, Applicants respectfully submit that claim 1 is in condition for allowance.

5. Independent claim 16 is patentable because neither Tada nor Fontaine disclose or suggest the claimed feature "inserting at least one code character into the electronic document to separate markup language from content" as recited in claim 16.

Applicants submit that claim 16 is allowable over the cited art for at least the following reasons. The combination of Tada and Fontaine fails to disclose, teach, or suggest at least the claimed element of "inserting at least one code character into the electronic document to separate markup language from content" as recited in step (a) of claim 16. The Final Office Action relies on Fontaine for teaching this claimed element. The Final Office Action on page 4 states:

> Fontaine teaches combining tags and information objects into a single nested structure in col. 4 line 31 – col. 5 line 44 and col. 5 line [sic]. Since, the nested structures of Fontaine logically define the order of a document and the informational objects contained within the document, the boundaries of the structure encapsulating the alias of Tada would have been separation variables separating the encode tag structure from the content.

However, Applicants respectfully disagree, as combining tags and information objects does not disclose the claimed feature of claim 16. The boundaries of a structure may not be

equated with the insertion of at least one code character. In fact, if the boundaries of Fontaine were discernable then the need for a separation variable would not be necessary.

In addition to and different from the boundaries argument discussed by the Examiner on Page 4 of the Final Office Action, the Examiner on pages 15-16 offers a specific example from Fontaine in which the Examiner alleges that the symbols %% used in the Fontaine preferred embodiment represent Applicants claimed separation variable. (We note that the Examiner must be refereeing to the claimed "at least one code character" as a separation variable is not claimed in independent claim 16). In particular, the Examiner states:

> Fontaine teaches an[d] [sic] example tag named "tagname" in col. 4 line 59. Since this tag cannot be distinguished from the text content, a separation variable is plainly shown also in col. 4 line 59. In the example of Fontaine, the variable %% is used to separate the tag from the text content. The separation variable enables Fontaine to identify the tag and invoke its definition which is described in col. 4 lines 49-55. Therefore, the Examiner maintains the position that Fontaine does teach the claimed separating a tag from content with a separation variable.

Applicants respectfully submit that the instructions symbolized by %% symbols used in Fontaine do not disclose, teach, or suggest the claimed feature of "inserting at least one code character into the electronic document to separate markup language from content." In Fontaine, an information object (tag) is recalled through instruction "%%tagname%%." The symbols %% used in Fontaine refers to a recall instruction and do not disclose, teach, or suggest Applicants claimed "at least one code character." Therefore for at least this reason, Applicants submit that claim 16 is patentable over the combination of Tada and Fontaine.

6. Independent claim 16 is patentable because neither Tada nor Fontaine disclose or suggest the claimed feature "replacing the tag with an alias, whereby the tag may be readily identified during run-time parsing of the document" as recited in claim 16.

The combination of Tada and Fontaine fails to teach at least the claimed element of "replacing the tag with an alias, whereby the tag may be readily identified during run-time parsing of the document" as recited in step (d) of claim 16. The Office Action relies on Tada, contending that "Tada teaches replacing the tag with an alias whereby the tag may be readily identified during run-time parsing of the document in col. 22 lines 6-20."

As discussed with respect to claim 1, at no point in these lines does Tada describe "replacing the tag with the alias" as recited in the claim. Rather, this portion of Tada merely describes obtaining an identification number, the identification number replacing logical structure discriminator (such as "a character string after the start character "<" and before the end character ">"). Col. 22, lines 10-11. Applicants respectfully submit that identification number in Tada is replacing content such as "a character string" and is not replacing a tag with the alias." Moreover, as stated above in Tada, a start tag is replaced by a control code "α". Applicant respectfully submits that replacing a start tag with control code does not disclose, teach, or suggest, the claimed feature of "replacing the tag with the alias." Therefore, for at least this additional reason, Applicants submit that independent claim 16 is patentable over the combination of Tada and Fontaine.

7. Independent claim 19 is patentable because neither Tada nor Fontaine disclose or suggest the claimed feature "at least one tag having encoded therein a predefined integer alias for the tag" recited in independent claim 19.

Claim 19 also stands rejected under Tada in view of Fontaine. The Final Office Action alleges that "Tada teaches a tag having encoded therein a predefined integer alias for the tag in col. 1 lines 7-13, col. 21 lines 50-64, and col. 22 lines 6-20" as recited in step (a) of claim 19. Final Office Action, page 5. Applicants respectfully traverse this rejection.

The cited portion of Tada does not describe "at least one tag having encoded therein a predefined integer alias for the tag." Rather, Tada describes obtaining a logical structure identification number corresponding to the logical structure discriminator. In fact, Tada describes that "in place of the start tag, a specific control code 'α' representative of the start of the logical structure is written and the obtained logical structure is written and the obtained logical structure identification number and logical structure length are written after the control code 'α' It is respectfully submitted that the Final Office Action does not address the claimed element of "at least one tag having encoded therein a predefined integer alias for the tag."

8. Independent claim 19 is patentable because neither Tada nor Fontaine disclose or suggest the claimed feature "a code separating the tag from the content portion, whereby the content and markup within the document may be readily parsed at run-time" recited in independent claim 19.

In addition, claim 19 is allowable for at least an additional reason. Claim 19 recited the claimed feature of "a code separating the tag from the content portion, whereby the content and markup within the document may be readily parsed at run-time." The combination of Tada and Fontaine fails to teach at least this claimed feature as discussed above with respect to claims 1 and 16. Therefore, for at least these reasons, Applicants submit that claim 19 is in condition for allowance.

9. Dependent claims 2, 5, 7, 9, and 17-18, and 20 are patentable distinguishable over Tada and Fontaine.

Claim 2 depends from independent claim 1 and is allowable for at least the same reasons as independent claim 1 and for the additional claimed features recited therein.

Claim 5 depends from independent claim 1 and is allowable for at least the same reasons as independent claim 1 and for the additional claimed features recited therein.

Claim 7 depends from independent claim 1 and is allowable for at least the same reasons as independent claim 1 and for the additional claimed features recited therein.

Claim 9 is based on claim 1 and is allowable for at least the same reasons as independent claim 1.

Claim 17 depends from independent claim 16 and is allowable for at least the same reasons as independent claim 1 and for the additional claimed features recited therein.

Claim 18 is based on claim 16 and is allowable for at least the same reasons as independent claim 1.

Claim 20 depends from independent claim 19 and is allowable for at least the same reasons as independent claim 1 and for the additional claimed features recited therein.

B. Claims 3, 4, 8, 22, and 23 are patentably distinguishable over Tada in view of Fontaine and further in view of Open eBook.

Claims 3, 4, and 8 ultimately depend from independent claim 1 and are patentable over the combination of Tada, Fontaine and Open eBook for at least the above reasons discussed with respect to independent claim 1.

Claims 22 and 23 ultimately depend from independent claim 19 and are patentable over the combination of Tada, Fontaine and Open eBook for at least the above reasons discussed with respect to independent claim 1.

C. Claim 6 is patentably distinguishable over Tada in view of Fontaine and further in view of Carus.

Claim 6 depends from independent claim 1 and is patentable over the combination of Tada, Fontaine, and Open eBook for at least the same reasons as independent claim 1.

D. Claims 10 and 11 are patentably distinguishable over Tada in view of Carus.

Claim 10 recites:

> A method for pre-computing an electronic document having markup language content comprising the steps of:
>
> (a) identifying a tag between a left and a right term within a document;
>
> (b) determining whether the tag is within a single word; and
>
> (c) if the left and right terms are not part of a single word, inserting a word break flag between the left and right term,
>
> whereby a word break may be readily identified during a run-time search operation.

Applicants submit that claim 10 is allowable over the cited art for at least the following reasons. The combination of Tada and Carus fails to teach at least the claimed element of "determining whether the tag is within a single word" as recited in step (b) of claim 10. The Final Office Action admits and Applicants agree that "Tada does not teach comparing a left and right term to determine if they are part of a single word." Final Office Action, page 10. The Final Office Action relies on Carus, contending that "Carus does teach comparing a left and right term to determine if they are part of a single word . . . in col. 2 line 62 – col. 3 line 31 and col. 5

lines 51-67." Final Office Action, page 10. Furthermore, the Final Office Action on page 19 states:

> Carus teaches identifying a position and determining whether it is between words or within a word in col. 5 lines 51-67. Therefore, it provides the teaching of determining whether a tag is within a single word."

The Applicants respectfully disagree because Carus describes a structure in which "[t]he associated character-transition tag identifies the existence of a concatenation between successive characters, a break between successive characters, or an unknown transition between successive characters." Carus, col. 3 lines 28-32. The determination of a break between successive characters does not disclose, teach, or suggest the claimed feature of "identifying a tag between a left and a right term within a document." Claim 10 provides a "method for pre-computing an electronic document having markup language content comprising the steps of . . . determining whether the tag is within a single word," whereas Carus describes a structure where tags themselves identify a concatenation, break, or transition. Therefore, for at least these reasons, Applicants submit that claim 10 is in condition for allowance. Dependent claim 11 which ultimately depends from claim 10 is allowable for at least the same reason as independent claim 10.

E. Claims 12 and 13 are patentably distinguishable over Tada.

Claims 12 and 13 are patentable because Open eBooks fails to disclose the claimed features of a) "determining whether the portion is to be displayed for viewing by a reading device" and b) "if the portion is not to be displayed for viewing, inserting a no search flag in association with the portion" as recited in independent claim 12.

Independent claim 12 recites the claimed elements:

> (b) determining whether the portion is to be displayed for viewing by a reading device; and
>
> (c) if the portion is not to be displayed for viewing, inserting a no search flag in association with the portion,
>
> whereby a no search field may be readily identified and skipped during a run-time linear search.

Applicants submit that claim 12 is allowable over the cited art for at least the following reasons. Tada fails to teach at least the claimed element of "if the portion is not to be displayed for viewing, inserting a no search flag in association with the portion, whereby a no search field may be readily identified and skipped during a run-time linear search" as recited in step (c) of claim 12. The Final Office Action contends that "Tada teaches in col. 22 line 24 – col. 23 line 24 inserting a control code, which is a flag, to form an encoded structure indicating whether the information contained within the tags should be searched or not." Final Office Action, page 11. However, Tada actually compares two identification numbers to determine if the "matching process skip step is executed," and when the numbers do not match, "the text . . . is not read and is discarded." Tada, col. 22 line 53 – col. 23 line 3. Therefore, Tada teaches that it is the comparison between identification numbers that indicates whether the text is read, not the flag.

In addition, claim 12 is allowable over Tada for at least one additional reason. The Final Office Action admits and Applicants agree that "Tada does not teach that the no search flag is conditionally inserted based on determining whether the portion is to be displayed for viewing by a reading device." Final Office Action, page 11. Rather, the Final Office Action contends that:

> It would have been obvious to one of ordinary skill in the art at the time the invention was made to have modified Tada to have created the claimed invention. It would have been obvious and desirable to have used the search exclusion technique of Tada to have excluded portions which are not to be displayed by a viewing device from searching. This would have corresponded to the goal of Tada of improving run-time search operations as described in col. 6 lines 30 – col. 7 line 20.

Applicants respectfully disagree. The Final Office Action discloses no teaching or suggestion for the modification of Tada to insert a no search flag. Rather, Tada teaches and suggests the aforementioned comparison method. Thus, Applicants submit that claim 12 is in condition for allowance for at least these reasons. Dependent claim 13 which ultimately depends from claim 12 is allowable for at least the same reason as independent claim 12.

F. Claims 14 and 15 are patentably distinguishable over Open eBook.

Claim 14 recites in relevant part:

> A method for pre-computing an electronic document having markup language content comprising the steps of:

(a) identifying a Uniform Resource Locator (URL) within a document;

(b) searching a manifest file for a file referenced by the URL; and

(c) if the file is identified in the manifest file with a reference string, replacing part of the URL with the reference string and a flag for the file,

whereby the file referenced by the URL may be readily accessed when selected during run-time.

Applicants submit that claim 14 is allowable over the cited art for at least the following reason. Open eBook does not disclose, teach, or suggest the claimed feature of "replacing *part* of the URL with the reference string and a flag for the file." (Emphasis added). Thus, Applicants submit that claim 14 is in condition for allowance for at least this additional reason. Dependent claim 15 which ultimately depends from claim 14 is allowable for at least the same reason as independent claim 14.

G. Claims 24-32 are patentably distinguishable over Open eBook in view of Tada.

Claim 24 recites the claimed features of "wherein the content file is pre-computed and encoded to minimize computational run-time requirements." The Final Office Action states and Applicants agree that Open eBook does not teach wherein the content file is pre-computed and encoded to minimize run-time requirements. Final Office Action, Page 13. The Final Office Action states that Tada teaches the advantage of decreased search time as a result of pre-computing and encoding the content file. In addition, the Final Office Action states:

> It would have been obvious and desirable to have used the content file pre-computing and encoding as taught by Tada to have enabled fast run-time search operations on a Open eBook, when is often implemented on a low power portable reading device.

Final Office Action, Page 13.

Applicants respectfully disagree. Neither reference discloses or suggests this motivation to combine. It is respectfully submitted that just because the teaching of different documents may be combined and implemented together, it may not then be inferred that such motivation existed at the time of the invention by those of ordinary skill in the art. Therefore, Applicants

submit that independent claim 24 is allowable. Claims 25-31 depend from claim 24 and are also allowable as being dependent on an allowable base claim and further in view of additional claimed features recited therein.

Claim 32 is also stands rejected under Open eBook in view of Tada. Claim 32 recites the claimed features of "forming a converted document. . . . ." The Final Office Action states and Applicants agree that "Open eBook does not teach converting a document in a first format by processing the document to pre-compute and encode the markup language within the document. . . ." Final Office Action, Page 14. The Final Office Action states that Tada teaches converting a document in a first format by processing the document to pre-compute and encode the markup language within the document. Final Office Action, Page 15. In addition, the Final Office Action states:

> It would have been obvious and desirable to have used the document and pre-computing and encoding as taught by Tada to have enabled fast run-time search operations on a Open eBook, when is often implemented on a low power portable reading device.
>
> Final Office Action, Page 15.

Applicants respectfully disagree. Neither reference discloses or suggests this motivation to combine. It respectfully submitted that it is impermissible to use the present application as a blueprint to combine references when the only suggestion can be found in the present application. It is respectfully submitted that just because the teaching of different documents may be combined and implemented together, it may not then be inferred that such motivation existed at the time of the invention by those of ordinary skill in the art. Therefore, for at least this reason, the combination of Open eBook and Tada fails to support an obviousness type rejection for independent claim 32. Claims 33-35 depend from independent claim 32 and are allowable for at least the reason discussed above with regards to independent claim 32.

VIII. Conclusion

The rejections contained in the Final Office Action of December 23, 2005 and the Advisory Action of March 10, 2006 should be reversed for at least the reasons recited above. Reversal of the rejections is requested.

Respectfully submitted,

Date: By: William J. Allen 51,393

William J. Allen
Registration ~~No.~~ 51,393
BANNER & WITCOFF, LTD.
10 South Wacker Drive
Suite 3000
Chicago, IL 60606-7407
Telephone: 312-463-5000
Facsimile: 312-463-5001

APPENDIX

CLAIMS INVOLVED IN THE APPEAL

1. A method for encoding an electronic document having markup language content, wherein the document includes at least one tag and an associated content, the method comprising the steps of:

(a) separating the tag from the content with a separation variable;

(b) replacing the tag with an alias, wherein the alias is a pre-defined representation for the tag; and

(c) inserting at least one flag within the tag to form an encode tag structure, wherein a first encoded document is formed.

2. The method for encoding of claim 1, wherein the step of replacing includes the step of replacing at least one attribute type within the tag with an attribute alias, wherein the attribute alias is a predefined representation for the attribute type.

3. The method for encoding of claim 1, further comprising the steps of:

(d) UTF-8 encoding the first encoded document to form a second encoded document.

4. The method for encoding of claim 3, further comprising the step of:

(e) compressing the second encoded document to form a compressed document.

5. The method for encoding of claim 1, wherein the step of inserting includes the step of inserting a position flag to indicate whether the tag is a start tag or an end tag.

6. The method for encoding of claim 1, wherein the step of inserting includes the step of inserting a word break flag between a left and right term of the associated content, whereby a word break may be readily identified during a run-time search operation.

7. The method for encoding of claim 1, wherein the step of inserting includes the step of inserting a no search flag in association with a portion of the content information, whereby a no search field may be readily identified and skipped during a run-time linear search.

8. The method for encoding of claim 1, further comprising the step of:

(d) replacing a URL within the content information with a reference string,

whereby the file referenced by the URL may be readily accessed when selected during run-time.

9. A computer-readable medium having computer-executable instructions for performing the steps recited in claim 1.

10. A method for pre-computing an electronic document having markup language content comprising the steps of:

(a) identifying a tag between a left and a right term within a document;

(b) determining whether the tag is within a single word; and

(c) if the left and right terms are not part of a single word, inserting a word break flag between the left and right term,

whereby a word break may be readily identified during a run-time search operation.

11. A computer-readable medium having computer-executable instructions for performing the steps recited in claim 10.

12. A method for pre-computing an electronic document having markup language content comprising the steps of:

(a) identifying a tag within a document associated with a portion of content;

(b) determining whether the portion is to be displayed for viewing by a reading device; and

(c) if the portion is not to be displayed for viewing, inserting a no search flag in association with the portion,

whereby a no search field may be readily identified and skipped during a run-time linear search.

13. A computer-readable medium having computer-executable instructions for performing the steps recited in claim 12.

14. A method for pre-computing an electronic document having markup language content comprising the steps of:

(a) identifying a Uniform Resource Locator (URL) within a document;

(b) searching a manifest file for a file referenced by the URL; and

(c) if the file is identified in the manifest file with a reference string, replacing part of the URL with the reference string and a flag for the file,

whereby the file referenced by the URL may be readily accessed when selected during run-time.

15. A computer-readable medium having computer-executable instructions for performing the steps recited in claim 14.

16. A method for encoding an electronic document comprising the steps of:

(a) inserting at least one code character into the electronic document to separate markup language from content;

(b) locating a tag within the electronic document associated with a portion of content;

(c) identifying a pre-defined integer alias for the tag; and

(d) replacing the tag with the alias,

whereby the tag may be readily identified during run-time parsing of the document.

17. The method of encoding of claim 16, further comprising the steps:

(e) locating an attribute type within the tag;

(f) identifying a pre-defined attribute alias for the attribute type; and

(g) replacing the attribute type with the attribute alias.

18. A computer readable medium having computer-executable instructions for performing the steps recited in claim 16.

19. A computer-readable medium having stored thereon a markup language document comprising in combination:

(a) at least one tag having encoded therein a predefined integer alias for the tag;

(b) an content portion associated with the tag;

(c) a code separating the tag from the content portion,

whereby the content and markup within the document may be readily parsed at run-time.

20. The computer-readable medium of claim 19, wherein the tag further includes at least one flag wherein the flag is selected from the group consisting of WORDBREAK, NOSEARCH, STARTTAG, and ENDTAG.

21. The computer-readable medium of claim 19, wherein the tag further includes at least one pre-defined attribute type alias.

22. The computer-readable medium of claim 19, wherein the markup language document is UTF-8 encoded.

23. The computer-readable medium of claim 22, wherein the markup language document is compressed.

24. A computer-readable medium having stored thereon an electronic book having a file format hierarchy comprising in combination:

(a) a root directory;

(b) a content subdirectory linked to the root directory, the content subdirectory having nested therein at least one linked content file providing content information relating to the electronic book, wherein the content file is pre-computed and encoded to minimize computational run-time requirements.

25. The electronic book of claim 24, further comprising:

(c) at least one link destination index file linked to the content file.

26. The electronic book of claim 24, further comprising:

(c) page break index providing an index of page break corresponding to the electronic book.

27. The electronic book of claim 24, further comprising:

(c) metadata file linked to the root directory and having information about the electronic book.

28. The electronic book of claim 24, further comprising:

(c) manifest file linked to the root directory providing a listing of the files in the content subdirectory relating to the electronic book.

29. The electronic book of claim 24, wherein the content database further includes at least one Cascading Style Sheets (CSS) file.

30. The electronic book of claim 24, further comprising:

(c) metadata file linked to the root directory and having information about the electronic book; and

31. The electronic book of claim 24, further comprising:

(c) a digital rights management database linked to the root database.

32. A method of converting an electronic document comprising markup language therein, the method comprising the steps of:

(a) receiving the document having a first format;

(b) processing the document to encode and pre-compute the markup language within the document; and

(c) forming a converted document, wherein the converted document has a file format hierarchy comprising in combination:

(i) a root directory; and

(ii) a content subdirectory linked to the root directory, the content subdirectory having nested therein at least one linked content file providing content information relating to the converted document.

33. The method of converting of claim 32, wherein the first format is an Open E-Book format.

34. The method of claim 32, wherein the document is an electronic book.

35. A computer-readable medium having computer-executable instructions for performing the steps recited in claim 32.

EVIDENCE APPENDIX

-- NONE --

RELATED PROCEEDINGS APPENDIX

-- NONE --